**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МОЭВМ**

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Обучение с подкреплением»**

**Тема: Реализация DQN для среды CartPole-v1**

| Студент гр. 0306 |  | Кирсанов Д.Э. |
| --- | --- | --- |
| Преподаватель |  | Глазунов С.А. |

**Цель работы**.

Реализация DQN для среды CartPole-v1. Исследование влияния различных параметров: архитектура сети, значения gamma и epsilon\_decay, влияние epsilon на скорость обучения

**Задание.**

1. Реализация DQN
2. Измените архитектуру нейросети (например, добавьте слои).
3. Попробуйте разные значения gamma и epsilon\_decay.
4. Проведите исследование как изначальное значение epsilon влияет на скорость обучения

**Выполнение работы.**

1. **Реализация DQN**

Основной созданной сущностью является DQNAgent, который оперирует моделью Нейронной Сети, отражённой в классе QNetwork, который в изначальном состоянии представлен на рисунке 1. Её входными параметрами являются Observation Space среды CartPole-1, описание которых представлено на рисунке 2, а на выходе численный показатель эффективности двух возможных действий: движение влево (индекс 0) или движения в право (индекс 1).

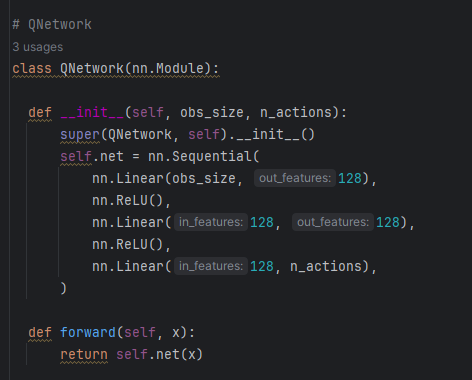


Рисунок 1 – Изначальная структура QNetwork

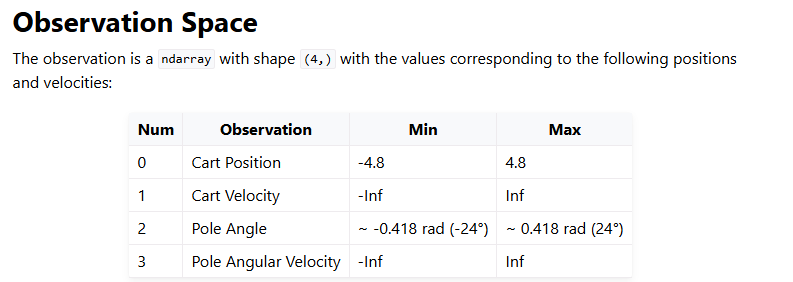


Рисунок 2 – Описание состояния среды CartPole-v1

QAgent помимо определяет следующие гипер параметры сети: оптимизатор (обновляет веса), значение gamma (степень влияние будущего состояния на выбор сети), batch\_size, стартовое значение epsilon (влияет на частоту выбора случайного действия), коэффициент уменьшения epsilon, пороговое значение epsilon. Их изначальное значение представлено на рисунке 3.

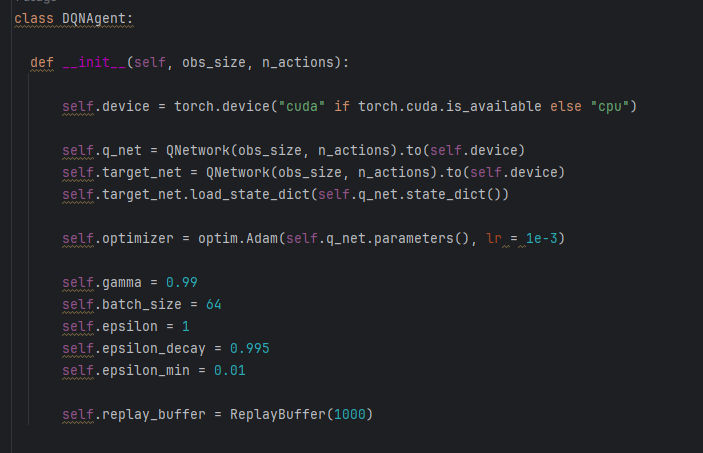


Рисунок 3 – Изначальное состояние гипер параметров сети

Все параметры изменяемые на дальнейших этапах работы представлены на рисунках выше. Остальной код, а именно тренировочный цикл, загрузчик данных ReplayBuffer, методы обновления весов, тренировки и обновления состояния целевой сети представлены в исходном скрипте DQN\_Routine.

1. Влияние изменения архитектуры сети на результаты

Результат 4 запусков при работе сети со структурой представленной ранее на рисунке 1 можно наблюдать на рисунках 4.

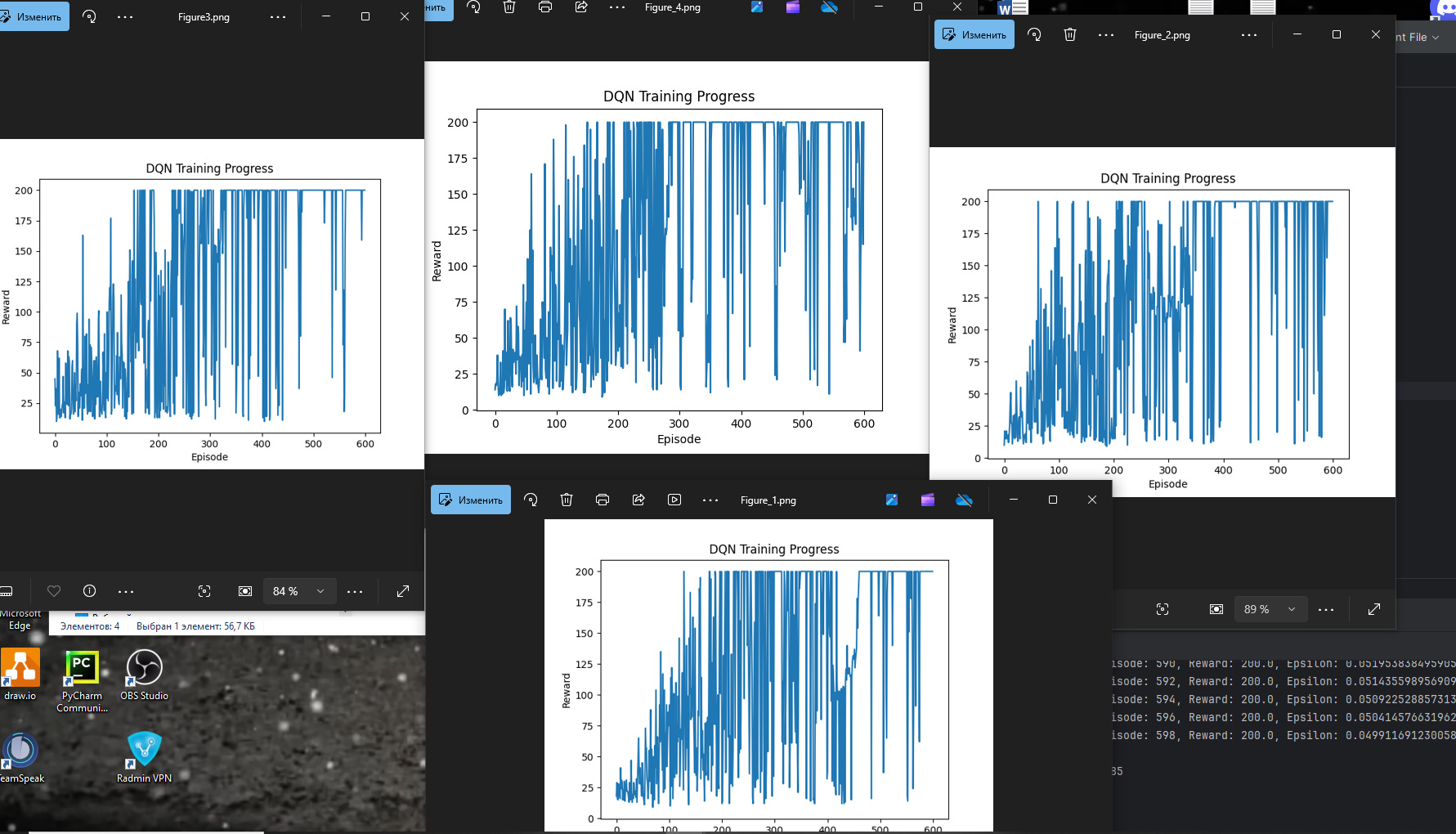


Рисунок 4 – 4 запуска с изначальной архитектурой

При такой архитектуре сеть достигает наибольшую награду, в районе 200, на 200 эпизоде. Сеть была изменена с добавлением дополнительных полносвязных слоев, в связке с ReLu, а также чуть уменьшено число параметров на каждом полносвязном слое, с целью посмотреть влияние изменения архитектуры на поведение сети. Текущая сеть представлена на рисунке 5, результат её работы на рисунке 6.

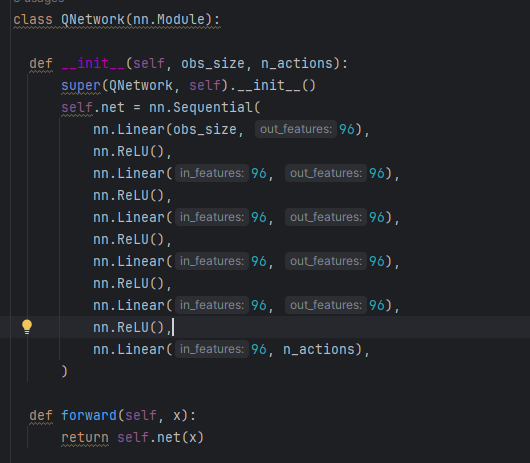


Рисунок 5 – Изменённая архитектура сети

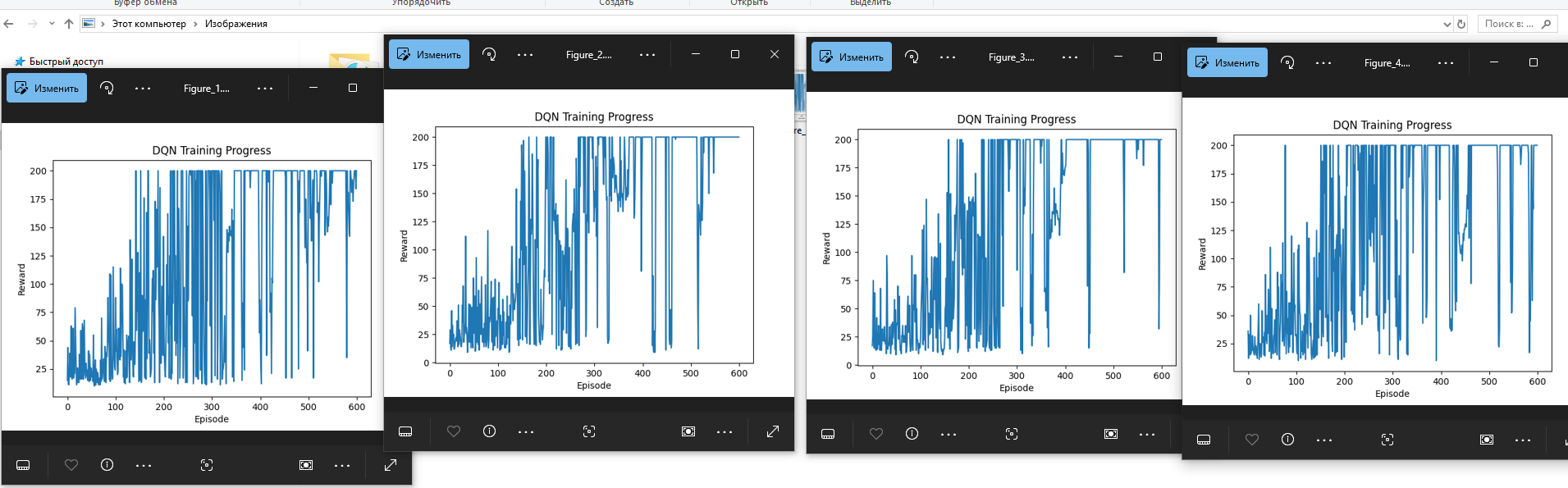


Рисунок 6 – 4 запуска с изменённой архитектурой

Хоть по результатам 4 запусков нельзя получить полную картину, но видно что после 200 эпизода, данная сеть чаще достигает максимальную награду. Этот вывод сделан, учитывая возможность удачного запуска этих 4 экспериментов. Более сильное видоизменение, например, добавление других видов слоев, требует более лучшего понимания DQN и дальнешей информации из лекций, а запуск большего числа экспериментов большого количества времени.

**2. Влияния gamma и epsilon**

**2.1 Изменение gamma**

Изменение gamma позволяет управлять предрасположенностью модели учитывать будущую награду. Меньшее значение gamma делает стратегию более жадной. Исходное значение gamma равное 0.99 было изменено на 0.5. Результат работы при данном значении параметра и на архитектуре сети из рисунка 5, представлено на рисунке 7.

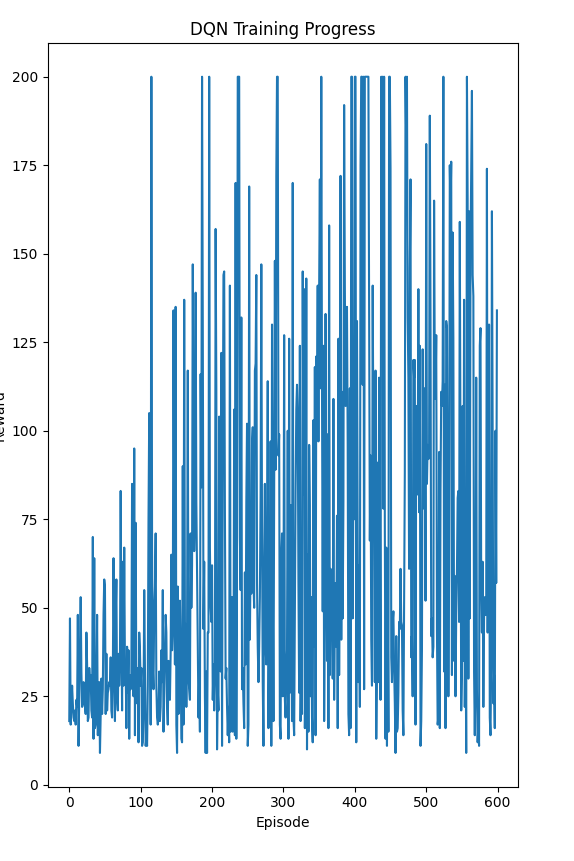


Рисунок 7 – Смещение к более жадной стратегии

Можно заметить если в прошлом случае, рисунок 6, были интервалы, где награда на протяжении нескольких эпизодов держалась на 200, то здесь эти интервалы значительно меньше. Можно сделать вывод, что применение более жадной стратегии не очень хороший метод ведения действий.

2.2 Изменение epsilon\_decay

epsilon\_decay не имеет смысла рассматривать в отдельности от значения epsilon, поэтому epsilon\_decay будет упомянот в пункте 3.

1. **Влияние epsilon на обучение**

Параметр epsilon влияет на предрасположенность сети делать случайные действия, что позволяет сети пробовать новые для себя действия и иметь информацию об их. epsilon\_decay позволяет менять во время обучения сети частоты появления этих случайных действий. Результат обучения со значением epsilon\_decay равным 0.7 представлен на рисунке 8.

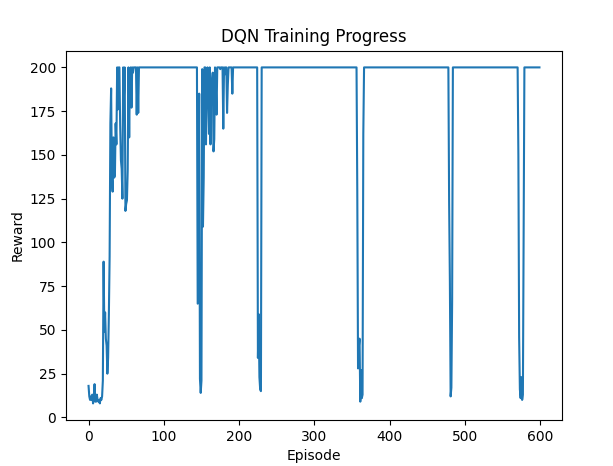


Рисунок 8 – Результат обучения при epsilon\_decay = 0.7

Из результатов видно, что меньшая частота появления случайных действий улучшила результат работы сети: награда в 200 появляется уже на 200 шаге, а интервалов с высокой наградой стало больше и сами они длиннее. Можно сделать вывод, что данная задача требует большего контроля и определенного шаблона, чем использование случайных действий

**Выводы.**

Была выполнена реализация DQN для среды CartPole-v1. Было проведено исследование влияния изменения некоторых параметров сети на её результат.